








Understanding value-driven delivery

Leading through change and challenges

Agile opportunities

-  **Video:** The evolution of Agile
5 min
-  **Reading:** The relationship between Agile and DevOps
20 min
-  **Reading:** Scaling Agile
20 min
-  **Practice Quiz:** Test your knowledge: Frameworks for scaling Agile
4 questions
-  **Video:** Jez: My thoughts on Agile
2 min
-  **Video:** Agile project management opportunities
5 min
-  **Ungraded Plugin:** Respond: Agile interview questions
30 min

Course review: Agile in practice

Optional: Uncovering job opportunities

Scaling Agile

In the preceding videos, we've covered how to run a Scrum team of up to nine team members. But what do you do if your team is larger than that? Or if the size of the product or solution is so large that multiple teams are required to do the work? In this reading, we will explore five frameworks that scale the Agile approach to address the needs of large initiatives or solutions: **Scaled Agile Framework (SAFe)**, **Scrum of Scrums**, **Large Scale Scrum (LeSS)**, **Disciplined Agile Delivery (DAD)**, and the **Spotify Model**.

Scaled Agile Framework (SAFe)

The most popular scaled framework is the **Scaled Agile Framework** [↗](#), or **SAFe**. SAFe is a Lean-Agile scaling framework that draws heavily on concepts from Kanban, Scrum, Extreme Programming (XP), DevOps, and Design Thinking methodologies. SAFe puts the goal of delivering value above all else—the first principle of SAFe is “take an economic view.” The framework organizes all work and teams into “Agile Release Trains” based on value streams; for example, sales. The SAFe framework is mature and provides detailed guidance on all elements of using SAFe, but some elements are more critical than others. Be sure to check back to the Agile principles and values in the manifesto to be sure you are preserving agility.

SAFe, like most Agile practices, is founded on a set of core values:

- **Alignment:** Synchronize the planning and execution of SAFe activities at all levels of the organization.
- **Built-in Quality:** Build quality into all stages of solution development.
- **Transparency:** Make execution activities visible at all levels to build trust among teams and across the organization.
- **Program Execution:** Focus on working systems and business outcomes.
- **Leadership:** Model the values and principles of SAFe.

Read this article to learn more about the [core values of SAFe](#) [↗](#).

Scrum of Scrums

Scrum of Scrums [↗](#) is a technique for integrating the work of multiple, smaller Scrum teams working on the same project or solution. Coordination among teams is critical to ensuring the deliverables from each team can be integrated into one larger, cohesive deliverable.

Scrum of Scrums involves the following elements:

- A group of at least 12 or more people divided into Scrum Teams of five to ten people each
- Scrum of Scrums meetings, which are held once a week, twice a week, or daily. These meetings follow the same format as a Daily Scrum meeting but focus on the Scrum team. In these meetings, you'll discuss questions like: “What did the team do yesterday? What problems occurred, if any, that are negatively affecting your team? What does your team want to accomplish before we meet again? Is your team blocked from moving forward on any tasks?”
- A **Scrum Master** or designated “ambassador” for each team that participates in the Scrum of Scrums meetings and a **Scrum of Scrums Master** who focuses on the overall Scrum process across multiple teams
- Sprint Planning, Sprint Review, and Sprint Retrospective meetings

Beyond these very basic guidelines, there is no official framework or methodology to implement Scrum of Scrums. Scrum of Scrums assumes that teams have a good working understanding of Scrum and are able to apply the scaling principles to how they work. Building on this knowledge, they design and iterate their own approach to coordinate multiple teams working on the same product.

Large-Scale Scrum (LeSS)

Large-Scale Scrum (LeSS) [↗](#) is a framework that aims to maximize the Scrum team's ability to deliver value and reduce waste in larger organizations. LeSS grew out of more than 600 experiments that expanded the practice of Scrum to larger groups.

LeSS includes ten principles for applying the value, elements, and overall purpose of Scrum across an organization. These principles were designed to create more customer- and collaboration-focused teams. LeSS teams prioritize learning, transparency, and customer needs. The ten LeSS principles are:

1. **Large-scale Scrum is Scrum:** Apply the values and principles of Scrum to a larger team.
2. **Empirical process control:** Inspect, adapt, and learn from experience to improve processes.
3. **Transparency:** Ensure clarity and accessibility across a project.
4. **More with less:** Create only necessary processes, roles, artifacts, and waste when scaling.
5. **Whole-product focus:** Think holistically about the product, making sure that all the parts serve the whole.
6. **Customer-centric:** Keep the customer's needs and values at the heart of your process.
7. **Continuous improvement towards perfection:** Improve the product—and your process—during every single Sprint.
8. **Systems thinking:** Think about the system as a whole; Don't get lost in the details.
9. **Lean thinking:** Seek continuous improvement, aim for perfection, and respect people.
10. **Queueing theory:** Embrace the Lean principles of “flow,” manage queue size,” and “minimize multitasking” to keep delivering value.

The LeSS toolkit provides two frameworks—one for up to about 50 people (called **Basic LeSS**) and one for 50–6000+ people (called **LeSS Huge**). More information on the LeSS frameworks can be found at [less.works](#) [↗](#).

Disciplined Agile Delivery (DAD)

Disciplined Agile Delivery (DAD) [↗](#) is a hybrid approach that combines the strategies from various Agile frameworks, including Kanban, LeSS, Lean Development, Extreme Programming, Agile Modeling, and more. DAD guides people through the process-related decisions that frameworks like SAFe and Scrum of Scrums leave open. DAD helps you develop a scaled Agile strategy based on context and desired outcomes.

DAD is organized into four “layers”:

1. **Foundations** discusses the principles, guidelines, Agile concepts, roles and team structure definitions, and Way of Working (WoW).
2. **Disciplined DevOps** ensures that solutions are delivered to customers effectively and safely, with data and security management always at the forefront.
3. **Value Streams** ensures that solutions are aligned with the organization's business strategy, connecting customers, sales, and portfolio management to the framework.
4. **Disciplined Agile Enterprise (DAE)** connects the industry marketplace with corporate governance and larger enterprise activities.

Project managers wishing to implement DAD can read more about the framework in this article: [Going Beyond Scrum](#) [↗](#).

The Spotify Model

Another approach you may encounter is the “Spotify Model,” which we discussed in a [previous reading](#). [↗](#) It is important to note that Spotify's model is not a true Agile framework. There is no standard guide on how to implement it. The model began as a description of how Spotify overcame the challenges of scaling Agile. By focusing their efforts on culture, team autonomy, communication, accountability, and quality, they increased their agility over time. Spotify's approach has had a huge impact on workflows and team structures across the tech world. Some of the key components include:

- **Squads:** Like Scrum teams, Squads are autonomous teams of 6–12 people working toward the same outcome. All Squads include a coach (similar to a Scrum Master) and a Product Owner.
- **Tribes:** When multiple Squads work on the same feature area, they form a Tribe of 40–150 people. Each Tribe has a Tribe Lead who fosters collaboration and coordination.
- **Chapters:** Squads may be autonomous, but specialists (e.g., JavaScript developers) should still align across an organization. Chapters establish best practices and, where necessary, set standards.
- **Guilds:** Any group of people interested in a certain topic can form a Guild, where people with shared interests can come together as a community.

While some organizations have had success with this model, be aware that it evolved from Spotify's already significant Agile experience. It is the product of extensive introspection and adaptation and draws heavily on the company's culture of trust, transparency, and autonomy. Therefore, the value of Spotify's approach to scaling is not in team names like Squads and Tribes but in how they developed practices that supported and served their organizational culture. To learn more about the Spotify Model, check out this [video](#) [↗](#) from Henrik Kniberg.

Best practices for scaling Agile

No matter which framework you choose, it's important to keep a few basic principles in mind:

- Treat scaling models like SAFe, Scrum of Scrums, LeSS, etc., as general frameworks, not instruction manuals.
- Different situations require different solutions. It's okay to mix and match elements from multiple frameworks, as long as you apply the principles and values of the Agile Manifesto.
- Don't try to scale without prior Agile experience. Going straight from Waterfall to scaled Agile can be risky without a knowledgeable guide.
- Finally, and most importantly, don't scale if it isn't necessary. The larger your team, the more complex and difficult your project becomes.

Key takeaway

Scaling Agile can be as simple as putting two Scrum teams together into a Scrum of Scrums configuration or as sophisticated as training an organization of thousands in the SAFe framework. When you have a large team or a big deliverable that requires multiple workstreams, think about how you can scale to suit your situation. Remember that you can modify SAFe, LeSS, and other scaled frameworks to meet the needs of each project. Make sure your team understands Agile principles before you try to scale since scaling inevitably introduces more waste and complexity.

Mark as completed

